基于跳数修正与 LM 优化的 DV-Hop 改进算法 *

高清源，徐曾春，胡　平

(南京工业大学 计算机科学与技术学院, 南京 211816)

摘　要：针对 DV-Hop 定位算法在节点分布不均匀的传感器网络拓扑环境下，跳数与跳距估计存在误差的问题，提出了一种名为 ILDV-Hop 的改进算法。首先基于信标节点间估计距离与真实距离的差值，提出了一种全网络的有效跳距。其次在信标节点与未知节点间多跳计算过程中增添了修正值，同时利用接收信号强度指示(RSSI)值优化单跳值的大小。最后采用列文伯格-马夸尔特算法估计未知节点的最优位置。仿真结果表明，与传统 DV-Hop 算法以及基于拟牛顿迭代的 DV-Hop 算法相比，ILDV-Hop 算法的定位误差分别减小了 23%与 10%左右，定位精度有了显著的提高。

关键词：DV-Hop 算法；有效跳距；跳数；列文伯格-马夸尔特算法

中图分类号：TP393　　doi: 10.3969/j.issn.1001-3695.2017.07.0664

Improved DV-Hop algorithm based on hop correction and LM optimization

Gao Qingyuan, Xu Zengchun, Hu Ping

(*Nanjing Tech University*, *School of Computer Science & Technology*, *Nanjing* 211816, *China*)

Abstract: In sensor network topology environment with uneven node distribution, running the DV-Hop localization algorithm often causes large errors of hop value. Considering the problem above, this paper proposed an improved localization algorithm named ILDV-Hop. Firstly, for the whole network, this algorithm calculated an effective hop size based on difference of actual and estimated distance between reference nodes. Then, with adding a correction value to the hops between unknown nodes and reference nodes, it also used the received signal strength indicator (RSSI) value to correct the distance of a single hop. Finally, the Levenberg–Marquardt algorithm was applied to estimate an optimize position for each sensor. Simulation results indicate that, compared with the DV-Hop algorithm and the DV-Hop algorithm based on Quasi-Newton, the ILDV-Hop algorithm has reduced the errors by about 23% and 10% respectively, and the localization performance has increased visibly.

Key Words: DV-Hop algorithm; effective hop size; hop count; Levenberg–Marquardt algorithm

0　引言

无线传感器网络通常由大量微型传感器节点组成，这些节点一般具有计算、通信、存储的能力[1]。传感器节点的主要任务为收集与聚合大量传感器网络环境下的遥感数据，并且将作为基站的汇聚节点作进一步处理[2]。倘若缺少由节点产生的包含于消息报告中的传感器位置信息，传感器网络则不能够正常工作，所以定位问题对于传感器网络的研究而言是一项巨大的挑战。许多定位方案假定只有小部分信标节点可通过手动布置或者自身携带 GPS 模块，使得拥有位置信息。但是这种方法不适合传感器网络，原因在于 GPS 系统有着功耗大、设备昂贵等缺陷[3]

无线传感器网络的定位算法大致根据是否要进行测量距离可分为两种类型：基于测距的定位算法与无须测距的定位算法，前者最常用的是基于到达时间（TOA）的定位[4]、基于到达角度（AOA）的定位[5]、基于接收信号强度指示（RSSI）的定位[6]，此类方法有利有弊，在提高了定位精度的同时却增加了硬件与人力的消耗。后者常用的为质心算法、DV-Hop 算法[7]、APIT 算法和 Amorphous 算法。此类算法无须使用外界设备，有着成本低等优势。其中 DV-Hop 算法是目前应用最为广泛的定位算法之一，其利用与未知节点最临近的信标节点的平均跳距进行未知节点的自定位，具有实现简单，成本低等优势。但是随着节点间跳数的累积，算法定位精度会大大降低，这是该算法主要存在的缺点。

针对 DV-Hop 算法在定位精度上出现的不足，国内外的学者们提出了各种改进方法：Kumar[8]等人提出了一种基于加权最小平方方法的 DV-Hop 改进算法，将 DV-Hop 定位算法第三阶段，未知节点自定位过程中的计算方法改为先相减再平方并加

基金项目：江苏省重点研发计划（社会发展）资助项目（BE2015697）

作者简介：高清源（1993-），男，江苏扬州人，硕士研究生，主要研究方向为无线传感器网络定位，物联网技术应用；徐曾春（1973-），女（通信作者），山东肥城人，工程师，硕士，主要研究方向为无线传感器网络，智能计算（gqyhunter@hotmail.com）；胡平（1962-），男，江苏南京人，教授，硕士，主要研究方向为人工智能及其应用、物联网技术．

权，通过误差修正项的比较可以得到该改进算法精度更高。但是此改进算法由于忽略了前阶段不同信标节点对未知节点的影响，使得定位优化效果不明显。Wen [9]等人提出了一种基于接收信号强度指示的改进算法，通过节点间收到的 RSSI 值对单跳数进行离散化处理，并且通过将节点间的距离作为权值来修正节点间跳数以达到减少定位误差的效果。该方法的弊端在于过多地增添了硬件功耗导致成本大幅度增加。Qiao 等人[10]提出了改进的 CNDV-Hop 算法，首先设定跳数阈值，对传感器网络中信标节点优选的同时对节点平均跳距进行修正，最后拟牛顿法通过迭代方式估算未知节点坐标。 Yu 等人[11]利用改进后的粒子群算法进行未知节点坐标自定位。但由于改进粒子群算法的复杂度偏大，使得该改进算法的计算量上升。

以上的研究工作在一定程度上都减小了定位误差，但是各自存在弊端。本文在传统 DV-Hop 定位算法基础上从误差的根源出发，提出了 ILDV-Hop 算法。首先重新定义全网平均跳距值，并用增添系数值的方法修正未知节点与信标节点跳数值，同时对节点间的单跳值进行离散化处理，最后阶段改用 LM 最优化算法通过迭代的方式进行未知节点自定位。

1　DV-Hop 算法

DV-Hop 作为一种典型的免测距型分布式算法，其定位过程大致可分为三个阶段。

a)每一个信标节点 i 利用距离矢量交换协议发送广播消息，广播的数据包中包含信标节点自身 ID（编号），坐标以及被初始化为 0 的跳数。这样网路中任意两个信标节点间的最小跳数值可被记录下。

b)在信标节点 i 获得其他信标节点的坐标和跳数值信息，可求得与其他信标节点间的平均每跳距离，这里表示为 $HopSize_i$。信标节点平均跳距的计算公式如下：

$$HopSize_i = \frac{\sum_{j=1,i\neq j}^{M}\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}{\sum_{j=1,i\neq j}^{M} hop_{ij}} \tag{1}$$

其中: M 是信标节点的数量， hop_{ij} 是信标节点 i 与 j 间的跳数值， (x_i,y_i) 与 (x_j,y_j) 分别代表信标节点 i 与 j 的坐标。

一旦跳距值 $HopSize_i$ 被计算出，信标节点会将其广播至整个网络。未知节点接收到所有信标节点的平均跳距值，再选取离其最近的信标节点的平均跳距值作为其计算标准。未知节点 s 与各信标节点 i 间的距离计算如下：

$$d_{si} = HopSize_i \times hop_{si} \tag{2}$$

c)未知节点利用信标节点的坐标与距离信息，并采用极大似然估计法来进行自定位。假设有 M 个信标节点被用于估计未知节点 s 的位置，可由第 2 阶段估算出的节点间距离得出以下系统方程：

$$\begin{cases}(x_s-x_1)^2+(y_s-y_1)^2=d_{s1}^{\ 2}\\(x_s-x_2)^2+(y_s-y_2)^2=d_{s2}^{\ 2}\\ \quad\vdots\\(x_s-x_M)^2+(y_s-y_M)^2=d_{sM}^{\ 2}\end{cases} \tag{3}$$

其中: (x_s,y_s) 为未知节点 s 的坐标。由于平方项的存在，式(3)属于非线性方程组。为使之线性化，用前 $M-1$ 个等式分别减去第 M 个等式，结果可重新写为 $AX=B$ 的形式。根据最小二乘法，未知节点 s 的坐标可由式(4)得到。

$$\hat{X}=(A^TA)^{-1}A^TB \tag{4}$$

其中：

$$A=-2\begin{bmatrix}x_1-x_M & y_1-y_M\\ \vdots & \vdots\\ x_{M-1}-x_M & y_{M-1}-y_M\end{bmatrix}$$

$$B=\begin{bmatrix}d_{s1}^{\ 2}-d_{sM}^{\ 2}-x_1^{\ 2}+x_M^{\ 2}-y_1^{\ 2}+y_M^{\ 2}\\ \vdots\\ d_{s(M-1)}^{\ \ 2}-d_{sM}^{\ 2}-x_{M-1}^{\ \ 2}+x_M^{\ 2}-y_{M-1}^{\ \ 2}+y_M^{\ 2}\end{bmatrix}$$

$$X=[x_s\quad y_s]^T$$ 。

2　DV-Hop 算法误差分析

通过上述 DV-Hop 算法定位过程的分析，可以得出该算法是通过网络中节点的连通度与距离矢量协议对未知节点进行定位，误差的产生主要有以下几个方面。

首先，在 DV-Hop 算法中，节点都是随机分布于传感器网络中的，一般由于网络拓扑结构导致节点分布的不均匀，所以信标节点间用曲折的斜线来计算平均跳距会使得误差存在，同时节点间跳数越多，误差越大。其次，原始算法中未知节点只将最邻近其自身的信标节点的平均跳距值作为其平均跳距计算的估计值，参照性十分有限。且若未知节点与多个信标节点间跳数值为 1 跳时，虽然远近相异，但是节点间估算的距离都是按照 1 跳来处理[12]，这往往会与真实情况不符。最后在未知节点自定位的计算中，一种非线性方程系统（如式(3)）被强行转换成了线性最小二乘问题。它将某一信标节点视为线性化工具，再利用例如求解正规方程组的直接法求解坐标，此计算过程忽略了线性化所带来的影响[13]，必然导致未知节点自定位会产生较大的迭代误差。

3　ILDV-Hop 定位算法

本章在传统 DV-Hop 定位算法的基础上，提出了名为 ILDV-Hop 的改进定位算法，主要针对 DV-Hop 定位算法第 a)b)阶段存在的误差进行改良。

3.1　信标节点间跳数修正

DV-Hop 算法的主要缺点是定位精度会受未知节点与信标节点间跳数的影响，当节点间的跳数增加或者出现节点分布不均匀的网络拓扑结构时，定位误差会增加。本节重新定义了信标节点每跳的平均距离值，具体步骤如下：

每一个信标节点首先利用已得到的跳数值与跳距值估算到其他信标节点的距离值，如式(5)所示。

$$est_dist_{ij} = HopSize_i \times hop_{ij} \quad (5)$$

其中：hop_{ij} 为信标节点 i 与 j 间的跳数值。然后信标节点通过式(6)计算出与其他信标节点间的真实距离。

$$true_dist_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (6)$$

接下来，每个信标节点利用得到的估计距离与真实距离，计算出距离误差。

$$error_{ij} = true_dist_{ij} - est_dist \quad (7)$$

其中：$error_{ij}$ 表示信标节点 i 与 j 间针对计算距离的误差估计。现提出一种利用误差估计得到的平均跳距有效值的概念，其用 $HopSize^i_{eff}$ 来表示，计算方法如下：

$$HopSize^i_{eff} = HopSize_i - ((\sum_{j=1,i\neq j}^{M} error / M) / \sum_{j=1,i\neq j}^{M} hop_{ij}) \quad (8)$$

最后，每一个信标节点将此有效跳距值随着洪泛过程广播至全网络。

3.2 未知节点跳数修正

3.2.1 单跳分级处理

当未知节点与信标节点间为 1 跳时，它们间估算出的距离值往往会出现较大误差。如图 1 所示，假设未知节点 A 的通信半径为 R ,在其通信范围中包含两个节点 B 与 C 。由于未知节点 A 在计算到节点 B 与 C 的距离时统一认为其相距的跳数皆为 1，这样导致距离估计不精确。图 1 中显然节点 C 距离 A 更近，未知节点 A 用全网平均跳距值计算出的与节点 C 的距离值会远大于它们的真实距离。因此本节提出一种基于 RSSI 值的节点间单跳修正方法。

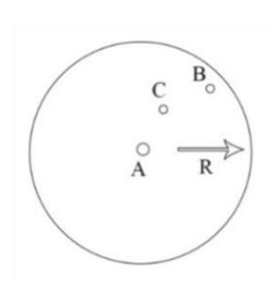


图 1　节点间位置关系图

RSSI 是一种通过信号传播过程中不断衰减的信号强度来估计节点间距离的技术[14],其在无线传感器网络中较为常见的为 Shadowing 模型，表达式如式(9)所示。

$$P_r(d) = P_r(d_0) - 10n\lg(\frac{d}{d_0}) + X_\sigma \quad (9)$$

其中：d 为两节点间的待估计距离，d_0 为参照距离，n 为可随外界因素改变的路径损耗指数。显然，节点间相距越远，接收到的信号强度值越小。

ILDV-Hop 改进算法中未知节点将接收到的信标节点的无线信号强度 $P_r(d)_{dB}$ 与其相距 $\frac{R}{2}$ 信标节点的信号强度 $P_r(\frac{R}{2})_{dB}$ 相比较。(其中本方法只包含比较信号强度的值，不考虑信号传播的范围。) 若 $P_r(d)_{dB}$ 的值大于 $P_r(\frac{R}{2})_{dB}$ 的值，则将未知节点与信标节点间跳数值记为 0.5 跳，否则将跳数值记为 1 跳。该方法将相隔为 1 跳的节点做跳数值分级处理，使得某些未知节点与信标节点间跳数值不再为整数，这种离散化过程有效地提高了跳数的可信性。

3.2.2 多跳加权处理

针对未知节点与信标节点间跳数值大于 1 跳的情况，为了控制通信功耗，不再使用 RSSI 值估算法，而是引入一定量修正值来优化跳数。权值计算方法如式(10)所示。

$$weight = k \cdot \frac{hop_{ij}}{HopSize^i_{eff}} \quad (10)$$

其中：hop_{ij} 为未知节点 i 与信标节点 j 间的跳数值，$HopSize^i_{eff}$ 为前阶段计算出的信标节点平均跳距有效值，k 为调整系数。当 hop_{ij} 值越大时，节点间计算出的距离值误差越大，证明修正值需要被设置得更大。未知节点与信标节点间的修正距离由式(11)表示。

$$corr_dist_{ij} = HopSize^i_{eff} \cdot (hop_{ij} + k\frac{hop_{ij}}{HopSize^i_{eff}}) \quad (11)$$

根据实验仿真数据，改进算法的定位误差与 k 值大小有着密切联系。例如,在节点通信半径在 25m，节点总数量为 100，且信标节点数量占总节点数量比例在 20%的条件下，参数 k 与定位误差值的关系图如图 2 所示。从图中可以得出，当 k 的值接近 1.2 时，节点定位误差降至最低点。在实际定位过程中，k 值大小可根据应用场景不同做特定选择。

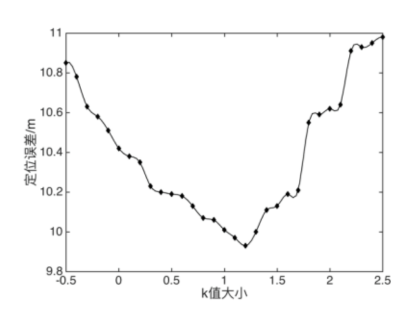


图 2　k 值大小与定位误差的关系

3.3 未知节点定位

在经典 DV-Hop 定位算法中，未知节点在接收到与信标节点跳段距离后，采用三边测量法或者极大似然估计法进行自定位。在此过程，一种非线性方程系统被强行转换成了线性最小二乘问题，导致计算误差存在。为了控制误差，必须改用非线性方法解决非线性问题，因此，ILDV-Hop 算法中引入优化概念，用列文伯格-马夸尔特（Levenberg–Marquardt，LM）优化算法[15]解决未知节点定位时非线性问题。

LM 优化算法是利用梯度求解最值的一种算法，其可用于解决非线性最小二乘得问题。LM 算法同时具有最优化算法中梯度下降法和高斯-牛顿法的特点，利用方向搜索的方式解决问

题的最优解。本文 ILDV-Hop 算法中未知节点坐标确定过程如下：首先，根据所有信标节点的坐标得到未知节点迭代初始坐标。再根据改进的 LM 算法初始值对未知节点坐标进行逐步迭代，最后得出满足要求的未知节点坐标最优解。每一次迭代中，非线性最小二乘法问题新的矢量方向会通过类似于梯度法的数学推导过程而被重新选择出。

在这里用 s_i 代表未知节点，其坐标为 $s_i=(x_i,y_i)$， a_j 代表信标节点，其坐标为 $a_j=(x_j,y_j)$。同时用 d_{ij} 表示未知节点与信标节点间的距离，则残差向量可由式(12)来表示：

$$d(s_i)=\begin{bmatrix} d_{i1}-\|s_i-a_1\| \\ d_{i2}-\|s_i-a_2\| \\ \vdots \\ d_{iM}-\|s_i-a_M\| \end{bmatrix} \tag{12}$$

因此，整个未知节点的自定位求解过程转换成为下式非线性最小二乘的问题[16]。

$$\min_{s_i} f(s_i)=\min_{s_i}(\frac{1}{2}d(s_i)^T d(s_i)) \tag{13}$$

现令 LM 改进算法中未知节点初始估计值为 s_0，且迭代过程中阻尼系数值为 μ_k，第 k 次迭代的增量正规方程中 $d(s_k)$ 的雅可比矩阵为 $J(s_k)$。 s_0，μ_k 与 $J(s_k)$ 值的计算公式分别由式(14)~(16)表示。

$$s_0=\frac{1}{M}\sum_{j=1}^{M}a_j \tag{14}$$

$$\mu_k=\rho\|J^T(s_k)d(s_k)\| \qquad \rho\in(0,1) \tag{15}$$

$$J(s_k)=\begin{bmatrix} \frac{\partial d(s_1^{(k)})}{\partial s_1} & \cdots & \frac{\partial d(s_1^{(k)})}{\partial s_M} \\ \vdots & & \vdots \\ \frac{\partial d(s_M^{(k)})}{\partial s_1} & \cdots & \frac{\partial d(s_M^{(k)})}{\partial s_M} \end{bmatrix} \tag{16}$$

这里选取迭代初始点的值为所有信标节点坐标值的算数平均值。LM 优化的独特之处在于增添了阻尼系数，对于 ILDV-Hop 算法中的阻尼系数 μ_k：当 ρ 值比较小趋于 0 时，LM 算法蜕变为高斯-牛顿法；相反当 ρ 值趋于 1 时，LM 算法则近似为梯度下降法。通常先设置一个比较小的 μ 值，当发现目标函数值反而增大时，将 μ 值增大使用梯度下降法快速寻找，然后再将 μ 值减小使用高斯-牛顿法进行寻找。

针对 ILDV-Hop 中针对未知节点自定位的 LM 算法，具体步骤如下所示：

a)计算未知节点迭代初始点 s_0，设定目标误差 ε，（ε 根据定位精度需求不同做相应设定），且令迭代次数 $k:=0$，步长因子 $v=10$。选取阻尼系数 μ 的初始迭代值 μ_0，本方法开始时的 μ_0 值需设定为较小，对应公式(14)中取 ρ =0.1 进行计算，即 $\mu_0=0.1\|J^T(s_0)d(s_0)\|$。

b)计算在第 k 次迭代中， $d(s_k)$ 的雅可比矩阵 $J(s_k)$，计算 $\overline{N_k}=J_k^T J_k+\mu_k I$（$I$ 为单位矩阵），然后构造增量正规方程 $\overline{N_k}\cdot\delta_k=J_k^T d(s_k)$。

c)首先求解步长 δ_k 的值，（$\delta_k=\overline{N_k}^{-1}J_k^T d(s_k)$），再做以下循环判断步骤：

(a)检验条件：当 $f(s_k+\delta_k)-f(s_k)<0$，此时若 $\|\delta_k\|<\varepsilon$，停止整个迭代过程并输出结果，此结果为未知节点最优坐标值；则若不满足，令 $s_{k+1}=s_k-\delta_k$，且 $\mu_{k+1}=\mu_k/v$，并且转到步骤 2。

(b) 如果 $f(s_k+\delta_k)-f(s_k)>0$，则令 $\mu_{k+1}=\mu_k\cdot v$，转到步骤 1 重新进行迭代至满足迭代的终止条件。

在此步骤中设置 $\|\delta_k\|<\varepsilon$ 为满足迭代的终止条件，且 ILDV-Hop 算法实验中的目标误差 ε 的设定值为 0.5。每次的迭代过程都是需要寻找一个合适的阻尼因子 μ，使得目标函数的值为下降的。

3.4　ILDV-Hop 定位算法过程

3.5　ILDV-Hop 算法总体流程如图 3 所示。

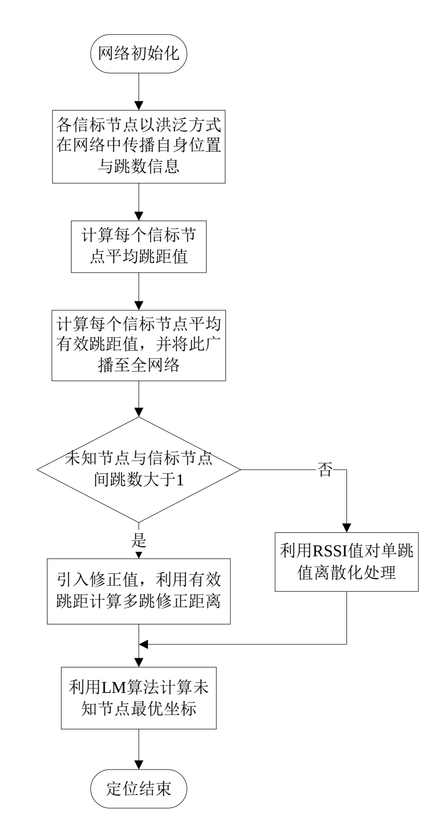


图 3　ILDV-Hop 定位算法流程图

4　实验分析与仿真

本节对 ILDV-Hop 算法与其他定位算法的性能进行分析对比。通过设置不同的场景，考察不同参数对于定位误差的影响。

4.1　仿真参数设定

为了验证算法的性能，本节使用 Matlab2014b 软件对各算法进行仿真对比。在仿真实验中，所有节点被随机部署于 100m $\times$ 100m 不含障碍干涉物（如空洞）的仿真区域内。所有信标节点被选择出随机分布于此区域内。节点的总数量设定为在 50~300 的范围内，节点的通信半径设置在 20 ~40 m 的范围内，信标节点的数量设置在 10~40 的范围内。

其余网络模型或仿真参数设置如下：

a)所有节点都静止（不存在移动锚节点）。

b)无线传输处于正常模式。

c)传感器节点具有同质性：节点在其通信与存储的能力上相似。

d)通信信道具有双向性。节点可以接收并且转发其他节点的信息。

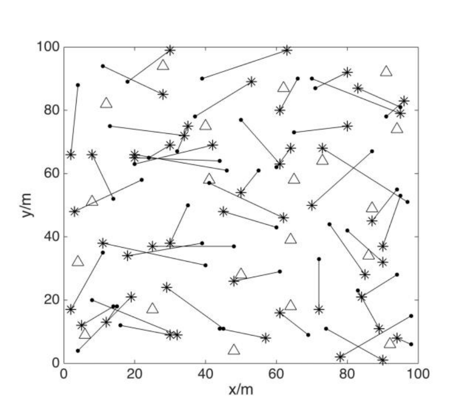
为了获得最优结果，实验在相同网络环境下进行 100 次仿真，最终结果是 100 次仿真的平均定位误差。未知节点的归一化平均定位误差（NALE）由式(17)表示。

$$NALE = \frac{\sum_{j=1}^{K}\sum_{i=1}^{N} dist(est_i, true_i)}{K \times N \times R} \tag{17}$$

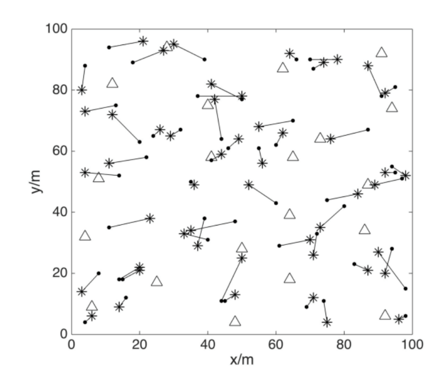
其中：N 为未知节点的数目，K 为仿真次数，取 100。$dist(est_i, true_i)$ 为传感器节点 i 的真实位置与估计位置的欧式距离，R 表示传感器节点的通信半径。

4.2　定位效果及误差分析

图 4(a)(b)分别仿真了当信标节点数目为 20，节点通信半径为 25 m 时，DV-Hop 算法与 ILDV-Hop 算法未知节点的真实位置与估计位置。此时信标节点数目占总节点的比例为 20%，未知节点的个数为 50。图中，“g”表示未知节点的真实位置，“*”表示未知节点的定位位置，“Δ”表示分布于网络中的信标节点。从 ILDV-Hop 算法定位效果图中可得其未知节点的真实位置与估计位置连线更短，即定位性能更优。



(a)DV-Hop 算法定位效果



(b) ILDV-Hop 算法定位效果

图 4　算法定位效果图

下面三组仿真将在设有不同参数的场景中进行，将 ILDV-Hop 算法的定位误差进行分析并与 DV-Hop 传统定位算法以及文献[10]中的 CNDV-Hop 改进算法相对比，使得三种算法的定位性能的优良更加明显。

场景 1　信标节点数量占总节点的比例恒定设置为 20%，节点的通信半径设置为 25m，总节点数量在 50 到 300 范围内变化。

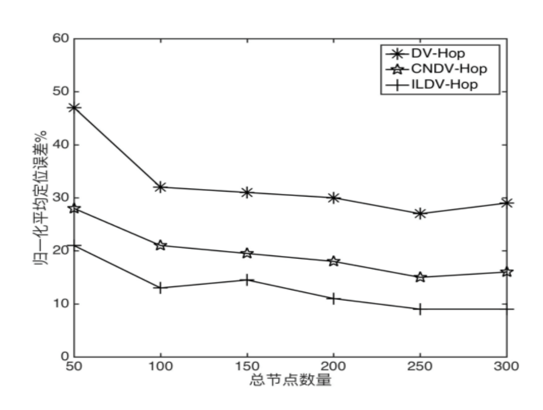


图 5　不同节点数量下算法定位误差比较

从图 5 可得到，随着节点总体数量的不断增加，节点间的连通度也相应增加，由于跳数与跳距信息估计更加准确，三种算法定位误差在减小。在节点总数量相同的条件下，ILDV-Hop 算法的定位误差要比 DV-Hop 算法与 CNDV-Hop 算法分别平均减小 24.5%和 10.2%左右。当节点总数目大于 200 时，ILDV-Hop 算法定位效果开始变得不显著，误差变化幅度相应减小。由于节点总数目的增加会引起网络结构变化，再加上算法局限性与仿真实验偶然性等多种复杂原因的存在，三种算法的定位误差折线不是理想状态下的单调下降，在节点数目增加的情况过程中出现了小幅度的上下波动，但从宏观上来看都呈不断减小并趋于稳定趋势。

场景 2　节点总数量恒定设置为 200，节点的通信半径设置为 25m，信标节点数量在 10 到 40 范围内变化。

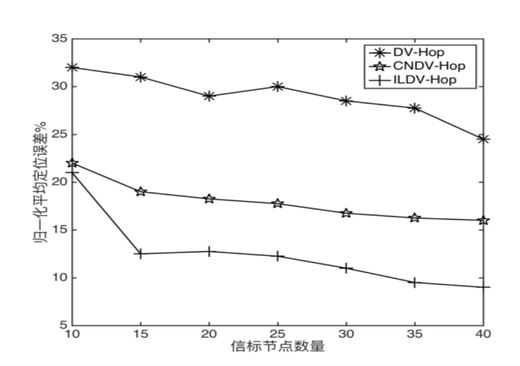


图 6　不同信标节点数量下算法定位误差比较

从图 6 可得到，当信标节点数量较少为 10 时，ILDV-Hop 算法的定位误差与 CNDV-Hop 算法差异不大，优势未能体现出。随着信标节点数目的增加，ILDV-Hop 算法的定位误差陡然减小，定位性能也都得到优化。当信标节点数目逐渐增多时，由于未知节点会获得更多的参照信息，三种算法的定位误差大致

成不断减小趋势。在信标节点数目相同的条件下，ILDV-Hop 算法的定位性能始终为最优的，其误差相比 DV-Hop 算法与 CNDV-Hop 算法分别平均减小 22.5%和 11.7%左右。同时，DV-Hop 算法误差折线由于其算法自身定位受干扰因素多，出现一定程度上下波动。

场景 3　节点总数量恒定设置为 200，信标节点数量占总节点数量的比例恒定设置为 20%。节点的通信半径在 20m 到 40m 范围内均匀增加。

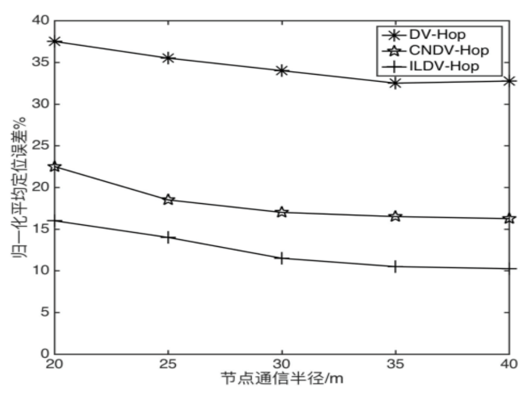


图 7　不同通信半径下算法定位误差比较

从图 7 可得到，随着节点通信半径不断增加，节点间的连通度受到影响，三种算法的误差都在小幅度减小且趋于稳定。在通信半径相同的条件下，ILDV-Hop 算法的定位误差始终为最小的，其相比 DV-Hop 算法与 CNDV-Hop 算法分别平均减小 23.6%和 8.8%左右。当节点通信半径大于 30m 时，三种算法的定位效果变得不显著，误差降低的幅度相应减小，且 ILDV-Hop 算法的定位精度受节点通信半径的影响最小。

5　结束语

本文针对传统 DV-Hop 定位算法在跳数依赖、线性化位置估计、跳距同一性等方面存在的问题，提出了名为 ILDV-Hop 的改进算法。该算法较传统定位算法做了以下改进：首先引入一种全网络的有效跳距，同时对信标节点与未知节点间的跳数值通过增添权值的方法进行修正，并且利用接收信号强度指示(RSSI)值对单跳值进行离散化处理，最后采用 LM 最优化算法代替极大似然法解决未知节点自定位过程中的非线性最小二乘问题。通过仿真结果可知，与传统 DV-Hop 定位算法相比，ILDV-Hop 算法在增加少量硬件功率消耗与算法计算量的情况下，很大程度提升了算法的精度。当然，如何进一步减小算法复杂度，同时提高在含空洞各向异性传感器网络的定位精度是本改进算法日后需要研究的问题。

参考文献：

[1] 曾园园. 无线传感器网络技术与应用 [M]. 北京：清华大学出版社, 2014: 16-28.

[2] 钱志鸿, 王义君. 面向物联网的无线传感器网络综述 [J]. 电子与信息学报, 2013, 35 (1): 215-227.

[3] 潘琢金, 刘文春, 罗振, 等. 无线传感器网络 DV-Hop 定位算法的改进 [J]. 计算机工程与设计, 2016, 37 (7): 1701-1704.

[4] Oguejiofor O S, Aniedu A N, Ejiofor H C, et al. Trilateration based localization algorithm for wireless sensor network [J]. International Journal of Science and Modern Engineering, 2013, 1 (10): 356-368.

[5] Naeimi S, Chow C O, Ishii H. Directional multi-hop clustering routing protocol for wireless sensor networks [J]. International Journal of Ad Hoc & Ubiquitous Computing, 2013, 14 (2): 123-134.

[6] Yang Xiang, Liu Meifeng, Deng Dengteng. An improved DV-Hop algorithm based on RSSI for WSN [C]// Proc of IEEE International Conference on Signal Processing, Communications and Computing. 2015: 1-5.

[7] Niculescu D, Nath B. DV based positioning in Ad hoc networks [J]. Telecommunication Systems, 2003, 22 (1): 267-280.

[8] Kumar S, Lobiyal D K. An advanced DV-hop localization algorithm for wireless sensor networks [J]. Wireless Personal Communications, 2013, 71 (2): 1365-1385.

[9] 温江涛, 范学敏, 吴希军. 基于 RSSI 跳数修正的 DV-Hop 改进算法 [J]. 传感技术学报, 2014, 27 (1): 113-117.

[10] 乔欣, 常飞, 丁恩杰, 等. 基于跳距修正的 WSN 拟牛顿迭代定位算法 [J]. 传感技术学报, 2014, 27 (6): 797-801.

[11] 于泉, 孙顺远, 徐保国, 等. 基于改进粒子群算法的无线传感器网络节点定位 [J]. 计算机应用, 2015, 35 (6): 1519-1522.

[12] 潘琢金, 刘文春, 罗振, 等. 无线传感器网络 DV-Hop 定位算法的改进 [J]. 计算机工程与设计, 2016, 37 (7): 1701-1704.

[13] Kaur A, Kumar P, Gupta G P. A novel DV-Hop algorithm based on Gauss-Newton method [C]// Proc of the 4th International Conference on Parallel, Distributed and Grid Computing. 2016: 625-629.

[14] Zhang Dengyin, Cui Guodong. A Union Node Localization Algorithm Based on RSSI and DV-Hop for WSNs [C]// Proc of International Conference on Instrumentation. 2013: 1094-1098.

[15] Dennis J E J, Schnabel R B. Numerical methods for unconstrained optimization and nonlinear equations [M]. 北京：科学出版社, 2009: 162-173.

[16] 祝强, 李少康, 徐臻. LM 算法求解大残差非线性最小二乘问题研究 [J]. 中国测试, 2016, 42 (3): 12-16.